Quantum Self-Organizing Map for Solving the Euclidean Traveling Salesman Problem

Rui XU
CIAD UR 7533
Université de Technologie de Belfort-Montbéliard
UTBM, F-90010, Belfort, France
rui.xu@utbm.fr

Jean-Charles Créput
CIAD UR7533
Université de Technologie de Belfort-Montbéliard
UTBM, F-90010, Belfort, France
jean-charles.creput@utbm.fr

***Abstract*—Hybrid quantum-classical algorithms are the subject of intensive research in quantum computing and specifically in the domains of Machine Learning and Approximate Optimization. Simple distance-based learning classifiers such as *k*NN or *k*-means classifiers have already been the subject of many proposals, but most of their quantum implementation only consider a very small number of classes/clusters and only use metrics suitable for normalized unit length quantum data. Very few approaches combine the need for Euclidean distance of unnormalized data with an efficient quantum computation implementation. Based on two *kNN* and *k-means* proposals from the literature, this paper proposes their adaptation to the design of a Quantum Self-Organizing Map in order to solve the NP-hard Traveling Salesman Optimization Problem (TSP) specifically. Hence, we evaluate the hybrid quantum-classical approach to Euclidean TSP instances of sizes from 52 to 127 cities and present the properties, advantages, and limitations of the quantum implementation in solving the TSP optimization problem.**

***Index Terms*—Quantum Machine Learning; Quantum Optimization; Self-Organizing Map; Traveling Salesman Problem**

I. Introduction

Quantum computing has been recognized as a highly promising method that could potentially increase the speed or improve the efficiency of specific computational tasks by utilizing the principles of quantum parallelism and interference [1]. Particularly, hybrid quantum-classical approach is appropriate for the noisy intermediate-scale quantum (NISQ) computing device [2], where fully quantum algorithms for NP-hard problems are limited by hardware. That's why quantum assistance to classical algorithm may provide advantages [3]. While a part of the computation is done classically, the most costly part should be processed by a Quantum Processing Unit to reach a real-life quantum advantage.

In the area of quantum machine learning, clustering and classification algorithms have been re-imagined for speed-ups by using quantum state superposition, interference and entanglement. The quantum *k*-means (q*k*-Means) algorithms [4] aim to partition data into k clusters using quantum subroutines, while quantum *k*NN (q*k*NN) [5]–[8] algorithms find k nearest neighbors in a dataset for classification. The quantum part for these two algorithms is naturally the closest point search, also called winner search, or best matching unit (BMU) search, based on some distance metric. Most of the time the metric operates on unit length preprocess data, with amplitude data encoding, with a very small number of classes. Very few approaches deal with Euclidean distance considering unnormalized data.

In this paper, we propose another application, that is, a Quantum Self-Organizing Map (QSOM) algorithm and its application to solve the NP-hard traveling salesman problem (TSP) [9], which will use a same winner search quantum part, as in the two papers [10], [11] for respectively *k*-means and *k*NN, and which deal with Euclidean distance estimation of unnormalized data. The techniques developed in q*k*-Means and q*k*NN, such as state preparation by amplitude encoding, distance evaluation via swap test circuits or related overlap tests, and quantum parallelism over data points, will be reused in our application as a first step in such a QSOM for TSP.

The Self-Organizing Map [12] was originally proposed by Teuvo Kohonen. It is a type of unsupervised neural network used for clustering and topological representation of data. It can be regarded as an extension of the *k*-means algorithm, adding relationships between clusters that constitute a low-dimensional grid or map, on which d-dimensional training data are projected. It is well known that it can be adapted to tackle TSP by using a one-dimensional grid of clusters, or neurons, that deploys among the data to approximate the tour [13]. To quantum-enhance SOM, we specifically target the BMU (best matching unit) search, which is essentially finding the nearest centroid (neuron) for a given city.

For example, Lazarev et al. (2020) [14] proposed a quantum-assisted self-organizing feature map, where their quantum circuit was only adapted for Hamming distances computation between binary feature vectors. Other proposals [15]–[17] also deal with very simple and restricted classification applications, with very few clusters. In our case, we propose an application to TSP that implies a large number of clusters with the same order of data size, and also a good estimate of Euclidean distance of the unnormalized data, which are requirements we do not find in the quantum literature at the date of writing. Also, we are looking for fast and effective quantum simulations.

In section II, we describe the classical SOM method for TSP, followed by the quantum-enhanced BMU search procedure in section III. We integrate and reformulate algorithms from

Poggiali et al. [10] and Zardini et al. [11] to suit the QSOM context, and provide circuit diagrams for the quantum BMU finder. In section IV, we design simulated experiments comparing classical SOM and QSOM on TSP instances, evaluating solution quality, runtime, and resource requirements. Through this comprehensive exploration, we demonstrate how quantum computing can interface with neural network heuristics to address complex optimization problems like the TSP.

II. Classical SOM for TSP

In this section, we present the Self-Organizing Map and its application to solving the TSP. Given N cities by their coordinates $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ in $\mathbb{R}^2$, the goal is to find a permutation (tour) π of $\{1, \ldots, N\}$ that minimizes the total Euclidean length:

$$L(\pi) = \sum_{i=1}^{N} \|\mathbf{x}_{\pi(i)} - \mathbf{x}_{\pi(i+1)}\|$$

with $\pi(N+1) \equiv \pi(1)$.

One of the first uses of SOM for solving the TSP was introduced by Angéniol et al. (1988) [13], who introduced a ring-based SOM in which neurons were arranged in a circular topology to approximate the optimal TSP tour. This structure allowed the neuron ring to iteratively adjust its positions toward the given cities, gradually forming a closed-loop path that closely follows an optimal TSP solution. This SOM approach is analogous to the elastic net [18] method and can handle very large instances where exact algorithms fail. Numerous improvements have been proposed [19] and a main property of SOM is its potential for massively parallel implementation as presented in [20].

To demonstrate the process of the Self-Organizing Map (SOM) approach, Figure 1 presents an illustrative example of the tour construction process for the TSP using a ring network. The visualization is based on the bier127 instance from the TSPLIB [21] benchmark set, and captures several stages of a simulation, highlighting the progressive adaptation of the network throughout the optimization process.

In the SOM approach, we define a set of M neurons (typically $M \approx N$ or a small multiple of N) arranged in a ring topology. Each neuron j has a weight (position vector) $\mathbf{w}_j \in \mathbb{R}^2$. The ring induces a neighbor relation: each neuron j has two immediate neighbors, $j-1$ and $j+1$ (indices mod M), and we can define a neighborhood function $h_{j,k}(t)$ that decreases with the ring distance between neuron j and neuron k modeled as a Gaussian.

The SOM algorithm initializes the learning weights $\{\mathbf{w}_j\}$ randomly. The SOM for TSP is displayed in Algorithm 1. SOM training typically runs for a fixed number of iterations. At each iteration, given a city point, a BMU search is performed. Once the winner neuron is found, the learning law modifies its weights and the weights of its immediate neighbors. Then the learning rates are slightly decreased. In our study, it is the BMU search that will be performed by quantum circuit, whereas weight updates remain classical.

Algorithm 1 SOM Algorithm for Solving TSP

```
Initialization: Initialize a set of neurons with weight vectors $\{\mathbf{w}_j\}$ randomly.
for $t = 0$ to $t_{\max}$ do
    Select Input City: Choose a random city $x_i$ from the dataset.
    Find Best Matching Unit (BMU): Identify the neuron closest to $x_i$:
        $k^* = \arg\min_j \|\mathbf{x}_i - \mathbf{w}_j(t)\|$
    Update Weights: Adjust the BMU and its neighbors:
        $\mathbf{w}_j(t+1) = \mathbf{w}_j(t) + \alpha(t) h_{j,k^*}(t)(\mathbf{x}_i - \mathbf{w}_j(t))$
    where:
      • $\alpha(t)$ is the learning rate, decreasing over time.
      • $\sigma(t)$ is the neighborhood radius, also decaying over time.
      • $h_{j,k^*}(t)$ is the neighborhood function typically Gaussian:
            $h_{j,k^*}(t) = \exp\left(-\frac{d_G(j,k^*)^2}{2\sigma(t)^2}\right)$
      • $d_G(j,k^*)$ represents the topological distance between neurons in the ring.
    Gradually reduce learning rate $\alpha(t)$ and radius $\sigma(t)$ of neighborhood
end for
Tour Extraction: Map each city to its nearest neuron and follow the neuron sequence to construct the tour.
```

III. Quantum Solution for the BMU Search

To incorporate quantum computing into the SOM, we focus on the winner search of finding the best matching unit. In classical SOM, finding the BMU for an input city $\mathbf{x}$ requires computing Euclidean distance $d(\mathbf{x}, \mathbf{w}_j)$ for all neurons j and picking the minimum. The goal is a quantum BMU finder that computes these distances in superposition and identifies the index j^* of the closest neuron with high probability. As a first step, in our paper, we study and reuse hybrid quantum k-means algorithms by Poggiali et al. [10] and quantum kNN of Zardini et al. [11] in our new context of SOM for optimization. Three increasing levels of quantum involvement are considered: (a) quantum distance calculation for one pair $(\mathbf{x}, \mathbf{w}_j)$ repeated for all j, (b) quantum parallel nearest-neighbor search for one input, and (c) quantum algorithm to quantum distance estimation for all pairs in parallel. Depending on the quantum BMU search strategy, two types of data encodings are employed. For qk-Means based method, Inverse Stereographic Projection (ISP encoding) maps 2-dimensional points to unit length normalized 3-dimensional points in the unit sphere, then each city and each neuron weight is mapped to a 3-dimensional normalized vector. For qkNN-based method, an extended feature encoding is used, where each city and each neuron weight is mapped in a F-dimensional feature vector,

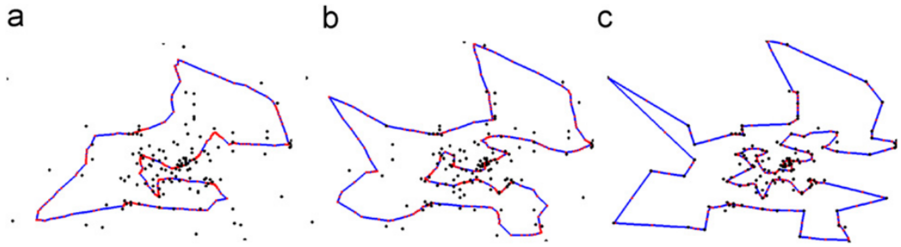

Fig. 1: SOM-based tour generation on the bier127 benchmark instance.

with $F = 2d+3$. This vector includes the original coordinates and the vector norm, thereby enriching the feature space to allow Euclidean distance estimation under quantum amplitude encoding. Next, the pair $(\vec{x}_j, \vec{w}_j)$ will denote the quantum encoding of the original pair $(\mathbf{x}, \mathbf{w}_j)$ in $\mathbb{R}^2$, note the index j that may be added if encoded vector $\vec{x}_j$ depends on j [11]. Globally, the quantum advantage lies with quantum parallel distance computation, while quantum data encoding rests on classical complexity.

A. Using $q_{1:1}$-k-Means for Quantum BMU Search

The $q_{1:1}$-k-Means variant [10] focuses on quantum distance calculation between a single input vector and a single centroid (neuron). Specifically, it estimates the squared Euclidean distance between two vectors using quantum interference within a dedicated circuit. A quantum circuit in Figure 2 prepares the superposition by using FF-QRAM data embedding [22], where $\hat{d} = \lceil \log(d) \rceil$ qubits ($d = 3$ for ISP encoding). For each pair $(\vec{x}, \vec{w}_j)$, a quantum state is constructed as:

$$|\psi\rangle = \frac{1}{\sqrt{2}} \left(|0\rangle_a |\vec{x}\rangle + |1\rangle_a |\vec{w}_j\rangle \right).$$

Here, $|\vec{x}\rangle$ and $|\vec{w}_j\rangle$ represent amplitude-encoded and normalized quantum states. The measurement probability is:

$$P(|0\rangle_a) = 1 - \frac{1}{4}\|\vec{x} - \vec{w}_j\|^2 \Rightarrow \|\vec{x} - \vec{w}_j\|^2 \quad = 4\left(1 - P(|0\rangle_a)\right).$$

This enables an estimation of the squared Euclidean distance. The BMU is then selected classically as:

$$j^* = \arg\min_j \left(1 - P_j(|0\rangle_a)\right).$$

Although this approach involves executing M quantum circuits per input, it is modular and relatively easy to implement on near-term quantum hardware.

B. Using $q_{1:M}$-k-Means for Quantum BMU Search

To augment the quantum computation part, we incorporate the quantum subroutine proposed in the $q_{1:M}$-k-Means algorithm [10] which itself, is an extension of the quantum kNN from [5], implemented via the circuit shown in Figure 3, where $\hat{d} = \lceil \log(d) \rceil$ and $\hat{M} = \lceil \log(M) \rceil$ are the number of qubits for data encoding. This approach enables a single quantum execution to simultaneously compare the input against all centroids in superposition. Assuming all vectors are normalized and amplitude-encoded, the entangled state is

$$|\Psi\rangle = \frac{1}{\sqrt{2M}} \sum_{j=1}^{M} \left(|0\rangle_a |\vec{x}\rangle + |1\rangle_a |\vec{w}_j\rangle \right) \otimes |j\rangle_\kappa,$$

where $|0\rangle_a$ is an ancilla qubit and $|j\rangle_\kappa$ denotes a label register encoding the index j. When a Hadamard gate is applied, we have:

$$|\Psi'\rangle = \frac{1}{2\sqrt{M}} \sum_{j=1}^{M} [|0\rangle_a(|\vec{x}\rangle + |\vec{w}_j\rangle) + |1\rangle_a(|\vec{x}\rangle - |\vec{w}_j\rangle)] \otimes |j\rangle_\kappa.$$

Post-selection is performed by measuring the ancilla in the $|0\rangle$ state and the FF-QRAM readout qubit in the $|1\rangle$ state. The surviving state is

$$|\Phi\rangle = \frac{1}{2\sqrt{Mp_a}} \sum_{j=1}^{M} (|\vec{x}\rangle + |\vec{w}_j\rangle) \otimes |j\rangle_\kappa,$$

where $p_a = \frac{1}{4M} \sum_{j=1}^{M} \|\vec{x} + \vec{w}_j\|^2$. Measuring the label register κ yields the index j with probability

$$P(j) = \frac{1}{4Mp_a} \|\vec{x} + \vec{w}_j\|^2 = \frac{1}{2Mp_a} \left(1 + \langle \vec{x}, \vec{w}_j \rangle\right),$$

which increases as the inner product $\langle \vec{x}, \vec{w}_j \rangle$ increases. Since $\|\vec{x} - \vec{w}_j\|^2 = 2 - 2\langle \vec{x}, \vec{w}_j \rangle$, maximizing $P(j)$ is equivalent to minimizing the Euclidean distance. Thus, the BMU is given by

$$j^* = \arg\max_j P(j) = \arg\min_j \|\vec{x} - \vec{w}_j\|^2.$$

In practice, this circuit is executed τ times to estimate the distribution $P(j)$ by sampling. The label with the highest observed frequency is selected as the BMU. This quantum approach replaces the classical linear scan with a constant-depth quantum circuit, significantly reducing the per-sample BMU search cost, especially when the number of neurons is large.

C. Using qkNN for Quantum BMU Search

In this method, we adopt the quantum k-nearest neighbors (qkNN) approach of [11] to estimate the squared Euclidean distances between a city vector $\mathbf{x} \in \mathbb{R}^2$ and a set of neuron weight vectors $\mathbf{w}_j \in \mathbb{R}^2, 1 \leq j \leq M$ in the SOM. Each vector is normalized and scaled to lie within the range $\left[-\frac{1}{2\sqrt{d}}, \frac{1}{2\sqrt{d}}\right]$, it is transformed to an extended feature vector and then encoded as a quantum state using amplitude encoding. The following entangled quantum state is prepared:

$$|\psi\rangle = |0\rangle \otimes \frac{1}{\sqrt{2}} \left(|0\rangle|\theta\rangle + |1\rangle|\eta\rangle\right),$$

where

$$|\theta\rangle = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} |j\rangle \sum_{i=0}^{F-1} w_{ji}|i\rangle, \quad |\eta\rangle = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} |j\rangle \sum_{i=0}^{F-1} x_{ji}|i\rangle,$$

with $\vec{w_j} = \{w_{ji}\}_{i=0,\ldots,F-1}$ is a weight (encoded) vector, $\vec{x_j} = \{x_{ji}\}_{i=0,\ldots,F-1}$ is a (encoded) city vector. A Hadamard gate is first applied to the ancilla qubit, followed by a CNOT operation as Bell-H circuit shown in Figure 4, where $I = \lceil \log(N) \rceil + \lceil \log(F) \rceil$, generating quantum interference. The probability of measuring the ancilla in state $|1\rangle$, given j is $P_j(1) = \frac{1}{2}\left(1 - \langle \vec{x}_j, \vec{w}_j \rangle\right)$, where $\langle \vec{x}_j, \vec{w}_j \rangle$ is the inner product between the two encoded vectors. Based on the encoding used, this inner product is related to the squared Euclidean distance by the identity

$$\|\mathbf{x} - \mathbf{w}_j\|^2 = \frac{3}{4} \langle \vec{x}_j, \vec{w}_j \rangle + \|\mathbf{x}\|^2.$$

Thus, we can estimate the distance $d_j = \|\mathbf{x} - \mathbf{w}_j\|^2$ from the measured probability $P_j(1)$, and select the best matching unit as

$$j^* = \arg\min_j \|\mathbf{x} - \mathbf{w}_j\|^2.$$

This method replaces classical distance computation with quantum amplitude interference, allowing a modular and low-depth quantum circuit for BMU search with competitive scaling in large SOMs.

IV. Experiments

We evaluate quantum-enhanced BMU search in SOM using city coordinate datasets from TSPLIB [21]. Each city is represented as an input vector $\mathbf{x} \in \mathbb{R}^2$. We develop the application of SOM for TSP, based on the two applications of Poggiali et al. [10] and Zardini te al. [11]. We formulate the BMU quantum part for evaluation in this new context of optimization. Five different methods are compared: a classical baseline using Euclidean distance (*Classical-SOM*), a quantum kNN method using exact inner product estimation of the encoded points via access to the statevector (*qkNN-statevector*), a sampling-based quantum kNN approach with many shots per query (*qkNN-local-simulation*), and the two quantum k-means variants, $q_{1:1}$-k-Means and $q_{1:M}$-k-Means, representing individual and superposition-based BMU computation respectively, but dealing with normalized data in ISP mode.

A. Setup

All experiments of the quantum BMU search algorithms used were adapted and executed using Python 3.12.9 and Qiskit 1.4.1, along with the GPU-accelerated backend provided by qiskit-aer-gpu 0.15.1. The experiments were run on a workstation with Ubuntu 24.04.2 LTS (x86_64), an Intel Core i7-8700 CPU, and an NVIDIA GeForce RTX 4060 Ti GPU. Quantum circuits were executed on Qiskit Aer simulators, using either the CPU or GPU, with adjustable shot counts.

B. Case Study: berlin52

We conducted a detailed case study on the TSPLIB instance berlin52, which contains 52 cities. We evaluate the effectiveness of quantum-enhanced BMU search in SOM-based TSP solving. Five different methods are compared.

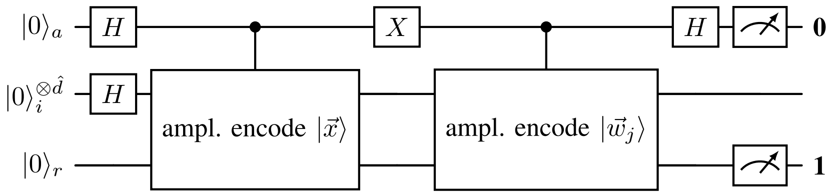
$|0\rangle_a$
$|0\rangle_i^{\otimes \hat{d}}$
$|0\rangle_r$
H
H
ampl. encode $|\vec{x}\rangle$
X
ampl. encode $|\vec{w}_j\rangle$
H
0
1

Fig. 2: $q_{1:1}$-k-Means: quantum Euclidean distance [10].

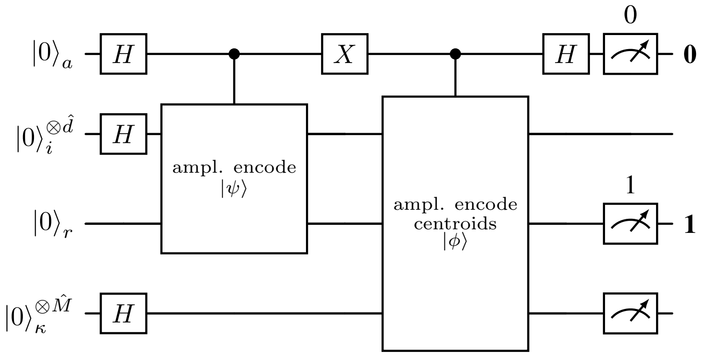
$|0\rangle_a$
$|0\rangle_i^{\otimes \hat{d}}$
$|0\rangle_r$
$|0\rangle_\kappa^{\otimes \hat{M}}$
H
H
H
ampl. encode $|\psi\rangle$
X
ampl. encode centroids $|\phi\rangle$
H
0
0
1
1

Fig. 3: $q_{1:M}$-k-Means: quantum kNN classifier [5], [10].

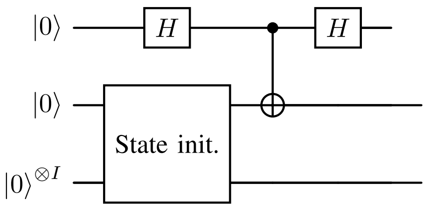

Fig. 4: Bell-H circuit [11]

Figure 5 presents the route length in training iterations. The classical and statevector methods show smooth convergence toward almost optimal solutions. However, The local simulation variant demonstrates greater variance as a result of sampling noise, which can be reduced through an increased number of shots. For the qk-Means methods, they also demonstrate convergence. Although, $q_{1:M}$-k-Means shows greater instability in the early stages of training, it could be caused by measurement ambiguity in the label register.

Final TSP routes after training are shown in Figure 6. All methods build the ring from an initial random state to suboptimal tours, by using classical and statevector methods, the most efficient routes can be achieved. While sampling-based quantum methods cause small detours or irregularities. $q_{1:M}$-k-Means is especially sensitive to the number of shots and the measurement distribution. To improve accuracy, shot counts were drastically increased according to the original implementations. In our context of SOM for TSP, we set 1,024,000 shots for qkNN_local_simulation. For $q_{1:M}$-k-Means, at least 532,480 shots were used, calculated as $\min(10 \cdot N \cdot 1024,\ 1{,}000{,}000)$ with $N = 52$.

These results confirm that quantum-enhanced BMU selection can match classical SOM performance when measurement resolution is sufficient. Practical implementations based on sampling can obtain satisfactory results by increasing the number of shots.

C. Scalability with Larger Instances

To evaluate the effectiveness of the different BMU selection strategies in SOM for TSP solving, we compare the final route lengths obtained by the five methods in four benchmark instances: berlin52, st70, kroA100, and bier127 of TSPLIB. Rather than comparing against global optima, we use the classical SOM as a practical reference.

Each method was executed three times on each instance using different random seeds, and we report the mean final route, and runtime in Table I, along with the relative percentage gap (GAP) with respect to the classical SOM solution. The GAP is computed as:

$$\text{GAP} = \frac{L_{\text{method}} - L_{\text{classical}}}{L_{\text{classical}}} \times 100,$$

where L_{method} and $L_{\text{classical}}$ denote the average route lengths of the quantum and classical methods, respectively.

As reported in Table I, the qkNN (statevector) method consistently performs close to the classical SOM, achieving route lengths within 1.5% across all instances and completing training in less than 40 seconds even on the largest instance bier127. This confirms the reliability of ideal quantum simulations in accurately replicating classical BMU decisions.

In contrast, the qkNN (local simulation) approach relies on 1,024,000 shots per input. It displays larger deviations, reaching up to 50% on bier127. The runtimes are moderate, ranging from approximately 1 minute on berlin52 to over 4 minutes on bier127. The observed performance drop and increased execution time are mainly attributed to sampling in quantum measurement, which becomes more significant as the problem size grows.

For the quantum clustering methods, the GAP and the training time increase with the instance size. $q_{1:1}$-k-Means maintains relatively good accuracy, with GAP ranging from 5.25% on berlin52 to 12.31% on bier127. But it requires longer runtimes, up to 1.5 hours for bier127. This is due to the need to construct and transpile a separate quantum circuit for each centroid comparison per input. Meanwhile, $q_{1:M}$-k-Means involves a more complex superposition circuit. Although, it requires only a single circuit per input, it leads to approximately 25% to 35% faster runtimes compared to $q_{1:1}$-k-Means. This improvement comes at the cost of higher GAP values, reaching up to 21.80%. It is worth noting that we observed a high transpilation time around 3 seconds for each transpilation.

Overall, the results highlight a fundamental trade-off between solution quality and computational efficiency: while statevector simulation offer both high precision and low cost, sampling-based methods introduce measurable deviations and significantly increased runtimes, especially when transpilation overhead is considered.

V. Conclusion

We have study the application of two types of quantum best matching unit search from two approaches of the literature in a new application of the SOM for solving the TSP problem. The implementation on Qiskit platform shows that we had to drastically augment the number of shots to achieve better Euclidean distance estimates in this context of application. Next steps will consists of optimizing the implementation in order to avoid the many transpilation operations with the help of parameterized circuits. Also, we will turn to analyze more elaborated circuits including Grover search for minimum findings maintaining the need for Euclidean distance evaluation of unormalized data.

References

[1] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, 2010.

[2] J. Preskill, “Quantum Computing in the NISQ era and beyond,” *Quantum*, vol. 2, p. 79, Aug. 2018. [Online]. Available: https://doi.org/10.22331/q-2018-08-06-79

[3] X. Ge, R.-B. Wu, and H. Rabitz, “The optimization landscape of hybrid quantum–classical algorithms: From quantum control to nisq applications,” *Annual Reviews in Control*, vol. 54, pp. 314–323, 2022. [Online]. Available: https://doi.org/10.1016/j.arcontrol.2022.06.001

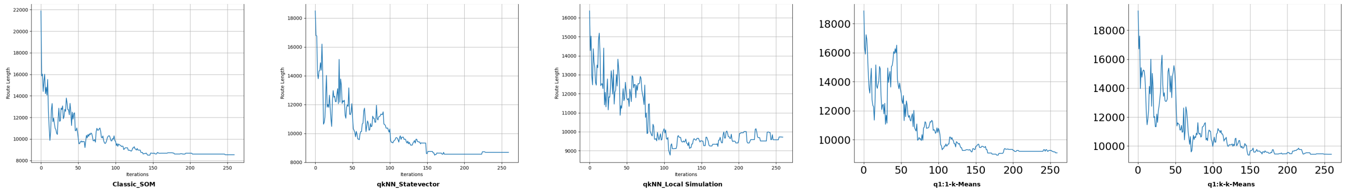

Fig. 5: Route length over training iterations for five BMU selection methods on berlin52

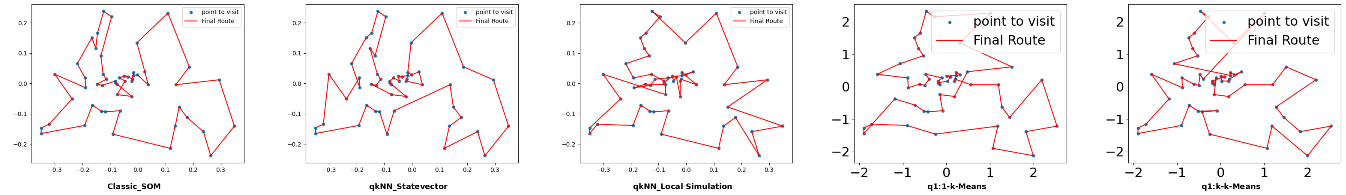

Fig. 6: Final TSP routes learned by the five methods on berlin52 after 260 iterations.

TABLE I: Average final route lengths with relative GAP (%) and total runtime (seconds) compared to Classical SOM baseline, computed over 3 runs.

Method	**berlin52**		**st70**		**kroA100**		**bier127**	
	Length (GAP)	Time (s)	Length (GAP)	Time (s)	Length (GAP)	Time (s)	Length (GAP)	Time (s)
Classical SOM	8604	-	733	-	24566	-	129238	-
q*k*NN Statevector	8665 (0.71%)	6	736 (0.41%)	16	24422 (–0.59%)	25	130676 (1.11%)	34
q*k*NN Local Simulation	10003 (16.27%)	63	823 (12.28%)	150	30139 (22.68%)	200	193445 (49.65%)	270
$q_{1:1}$-*k*-Means	9056 (5.25%)	960	778 (6.14%)	1680	26899 (9.50%)	3480	145126 (12.31%)	5460
$q_{1:M}$-*k*-Means	9658 (12.26%)	540	844 (15.14%)	1260	28742 (17.06%)	2580	157346 (21.80%)	4140

[4] I. Kerenidis, J. Landman, A. Luongo, and A. Prakash, *q-means: a quantum algorithm for unsupervised machine learning*. Red Hook, NY, USA: Curran Associates Inc., 2019.

[5] M. Schuld, M. Fingerhuth, and F. Petruccione, "Implementing a distance-based classifier with a quantum interference circuit," *EPL (Europhysics Letters)*, vol. 119, no. 6, p. 60002, 2017.

[6] A. Afham, A. Basheer, and S. K. Goyal, "Quantum *k*-nearest neighbors algorithm," *arXiv preprint arXiv:2003.09187*, 2020. [Online]. Available: https://arxiv.org/abs/2003.09187

[7] J. Li, S. Lin, K. Yu, and G. Guo, "Quantum k-nearest neighbor classification algorithm based on hamming distance," *Quantum Information Processing*, vol. 21, no. 1, p. 18, 2022. [Online]. Available: https://link.springer.com/article/10.1007/s11128-021-03361-0

[8] A. V. Jasso, A. Modi, R. Ferrara, C. Deppe, J. Nötzel, F. Fung, and M. Schädler, "Quantum and quantum-inspired stereographic k nearest-neighbour clustering," *Entropy*, vol. 25, no. 9, p. 1361, 2023. [Online]. Available: https://www.mdpi.com/1099-4300/25/9/1361

[9] C. H. Papadimitriou, "The euclidean traveling salesman problem is np-complete," *Theor. Comput. Sci.*, vol. 4, pp. 237–244, 1977. [Online]. Available: https://api.semanticscholar.org/CorpusID:19997679

[10] A. Poggiali, A. Berti, A. Bernasconi, G. M. Del Corso, and R. Guidotti, "Quantum clustering with k-means: a hybrid approach," *Theoretical Computer Science*, vol. 999, p. 114466, 2024. [Online]. Available: https://doi.org/10.1016/j.tcs.2024.114466

[11] E. Zardini, E. Blanzieri, and D. Pastorello, "A quantum k-nearest neighbors algorithm based on the euclidean distance estimation," *Quantum Machine Intelligence*, vol. 5, no. 1, pp. 1–15, 2023. [Online]. Available: https://doi.org/10.1007/s42484-023-00072-0

[12] T. Kohonen, "Self-organized formation of topologically correct feature maps," *Biological Cybernetics*, vol. 43, no. 1, pp. 59–69, 1982.

[13] B. Angéniol, G. de La Croix Vaubois, and J.-Y. L. Texier, "Self-organizing feature maps and the travelling salesman problem," *Neural Networks*, vol. 1, pp. 289–293, 1988. [Online]. Available: https://api.semanticscholar.org/CorpusID:27563425

[14] I. D. Lazarev, M. Narozniak, T. Byrnes, and A. N. Pyrkov, "Hybrid quantum-classical unsupervised data clustering based on the self-organizing feature map," *Physical Review A*, vol. 111, no. 1, p. 012416, 2025. [Online]. Available: https://doi.org/10.1103/PhysRevA.111.012416

[15] S. Garavaglia, "A quantum-inspired self-organizing map (qisom)," in *Proceedings of the 2002 International Joint Conference on Neural Networks. IJCNN'02 (Cat. No.02CH37290)*, vol. 2, 2002, pp. 1779–1784 vol.2.

[16] P. Li, Y. Chai, M. Cen, N. Liu, and Y. Qiu, "A quantum self-organizing mapping neural network," in *Proceedings of the 32nd Chinese Control Conference*, 2013, pp. 3264–3268.

[17] Z. Ye, K. Yu, G.-D. Guo, and S. Lin, "Quantum self-organizing feature mapping neural network algorithm based on grover search algorithm," *Physica A: Statistical Mechanics and its Applications*, vol. 639, p. 129690, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0378437124001997

[18] R. Durbin and D. Willshaw, "An analogue approach to the travelling salesman problem using an elastic net method," *Nature*, vol. 326, no. 6114, pp. 689–691, 1987.

[19] J.-C. Créput and A. Koukam, "A memetic neural network for the euclidean traveling salesman problem," *Neurocomputing*, vol. 72, no. 4, pp. 1250–1264, 2009, brain Inspired Cognitive Systems (BICS 2006) / Interplay Between Natural and Artificial Computation (IWINAC 2007). [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231208001434

[20] H. Wang, N. Zhang, and J.-C. Créput, "A massively parallel neural network approach to large-scale euclidean traveling salesman problems," *Neurocomputing*, vol. 240, pp. 137–151, 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231217303326

[21] G. Reinelt, "Tsplib—a traveling salesman problem library," *ORSA Journal on Computing*, vol. 3, no. 4, pp. 376–384, 1991.

[22] D. K. Park, F. Petruccione, and J.-K. K. Rhee, "Circuit-based quantum random access memory for classical data," *Scientific Reports*, vol. 9, 2019. [Online]. Available: https://api.semanticscholar.org/CorpusID:71715210